BAB 2
LANDASAN TEORI

2.1 Penyakit Kanker Paru

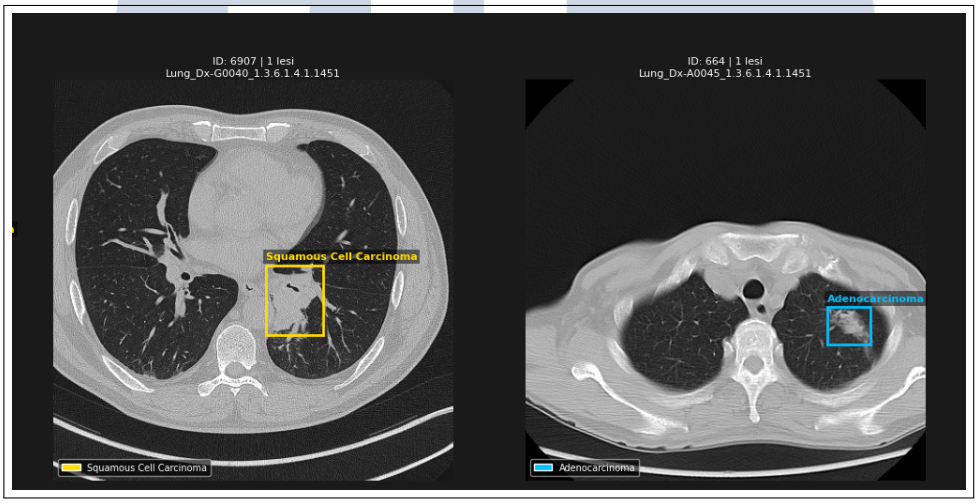
Kanker paru merupakan salah satu penyebab utama insidensi dan kematian akibat penyakit kanker di seluruh dunia [1]. Sebagian kasus kanker paru termasuk dalam kelompok *non-small cell lung cancer* (NSCLC) yang memiliki variasi yang beragam [3]. Perubahan pola gaya hidup, paparan lingkungan, hingga profil histologis dan genetik menjadi penentu utama dalam perkembangan penyakit ini [10]. Berdasarkan klasifikasi histologisnya, NSCLC secara umum terbagi ke dalam dua jenis berikut.

1. *Lung adenocarcinoma* (LUAD)
   Jenis kanker paru ini merupakan jenis kanker paru yang paling mendominasi secara global pada pasien pria maupun wanita. Peningkatan risiko perkembangan LUAD memiliki hubungan kausal yang kuat dengan paparan polusi udara partikulat di lingkungan [2]. Secara klinis, lesi tumor LUAD sering kali tumbuh pada area perifer atau bagian tepi paru-paru, dan umumnya memiliki ukuran tumor rata-rata yang lebih kecil saat pertama kali terdiagnosis dibandingkan jenis LUSC [11].

2. *Lung Squamous cell carcinoma* (LUSC)
   Jenis kanker paru ini menempati urutan insidensi kedua terbanyak setelah LUAD [2]. Perkembangan LUSC memiliki korelasi yang sangat kuat dengan pasien pria, riwayat kebiasaan merokok, serta tingginya konsumsi alkohol. Pasien dengan jenis LUSC cenderung menunjukkan gejala klinis yang lebih berat seperti batuk parah, demam, produksi dahak berlebih, serta memiliki tingkat kerentanan yang lebih tinggi terhadap infeksi bakteri dan jamur sekunder [3]. Berbeda dengan LUAD, lesi LUSC umumnya tumbuh pada area sentral di dekat bronkus utama, memiliki ukuran tumor yang lebih besar, dan dihubungkan dengan tingkat keganasan atau stadium klinis yang cenderung lebih tinggi saat didiagnosis [3].

Diferensiasi LUAD dan LUSC pada citra CT scan sangat bergantung pada peninjauan fitur semantik radiologis [12]. Secara visual, lesi LUAD umumnya ditandai oleh kemunculan *ground-glass opacity*, *air-bronchogram*, nodul satelit,

batas lobulasi yang dalam (*deep lobulated margins*), penebalan vaskular, serta penarikan pleura. Sebaliknya, jenis LUSC memiliki ciri khas berupa kavitasi akibat nekrosis jaringan dan batas lesi yang cenderung lebih halus (*smooth margins*) [12]. Kompleksitas dan variasi karakteristik visual inilah yang menuntut penggunaan arsitektur pengekstraksi fitur mutakhir untuk memfasilitasi klasifikasi secara otomatis [13]. Perbedaan bentuk dan lokasi jenis *lung adenocarcinoma* (LUAD) dan *lung squamous cell carcinoma* (LUSC) pada citra CT scan dapat dilihat pada Gambar 2.1.

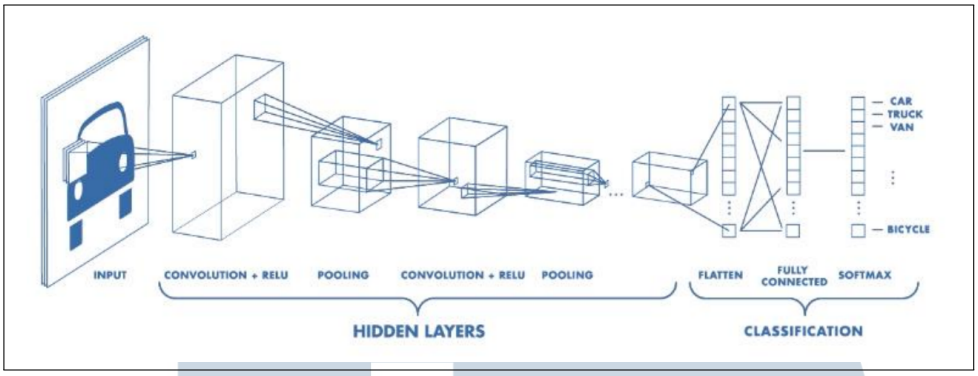


Gambar 2.1. Bentuk Jenis Kanker Paru LUAD dan LUSC pada Citra CT Scan

Sumber: [9]

2.2 Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan salah satu algoritma deep learning yang paling banyak digunakan dan telah menjadi jaringan saraf paling representatif dalam bidang *deep learning* [14]. Arsitektur CNN pada dasarnya terinspirasi dari struktur dan fungsi korteks visual manusia, sehingga desainnya menyerupai pola koneksi neuron pada otak manusia [14]. Secara umum, CNN tersusun atas empat jenis lapisan utama, yaitu lapisan konvolusi (*convolutional layer*), lapisan *pooling*, lapisan *fully connected*, dan lapisan non-linearitas atau fungsi aktivasi [14]. Detail Arsitektur CNN dapat dilihat pada Gambar 2.2.



Gambar 2.2. Arsitektur CNN

Sumber: [14]

2.2.1 Lapisan Konvolusi

Lapisan konvolusi merupakan komponen inti dari sebuah CNN yang berfungsi untuk mengekstraksi fitur dasar dari input gambar menggunakan filter (*kernel*) [14, 15]. Filter tersebut berupa matriks berukuran kecil yang digeser secara bertahap ke seluruh bagian input gambar. Pada setiap posisi pergeseran, dihitung hasil *dot product* antara nilai-nilai pada filter dengan nilai piksel pada area gambar yang sedang dilewati filter tersebut, sehingga menghasilkan sebuah *feature map* [14, 15].

Terdapat tiga parameter penting yang menentukan karakteristik lapisan konvolusi, yaitu ukuran filter (*kernel size*), panjang langkah pergeseran filter (*stride*), dan padding. Ukuran filter menentukan seberapa besar area citra yang diproses dalam satu kali operasi konvolusi sekaligus memengaruhi tingkat kompleksitas fitur yang dapat ditangkap [15]. *Stride* menentukan jarak pergeseran filter setiap kali berpindah posisi. Semakin besar nilai *stride*, semakin kecil resolusi *feature map* yang dihasilkan. *Padding* adalah penambahan bingkai bernilai nol di sekeliling input gambar yang berfungsi menjaga dimensi spasial citra selama proses konvolusi berlangsung [14, 15].

2.2.2 Lapisan *Pooling*

Lapisan *pooling* berfungsi menggabungkan hasil dari dua lapisan konvolusi berurutan dengan cara melakukan *down-sampling* terhadap *feature map*, sehingga jumlah parameter dan beban komputasi jaringan dapat dikurangi [14]. Operasi pooling yang umum digunakan adalah pengambilan *max pooling* atau *average*

pooling dari suatu area tertentu pada feature map [15]. Selain mengurangi beban konputasi, *pooling* juga membantu mencegah *overfitting* dan membuat jaringan lebih tahap terhadap pergeseran posisi kecil pada input gambar [14, 15].

2.2.3 Lapisan *Fully Connected*

Lapisan *fully connected*, yang juga dikenal sebagai *convolutional output layer*, menghubungkan setiap neuron pada lapisan sebelumnya dengan setiap neuron pada lapisan tersebut, sehingga strukturnya menyerupai jaringan saraf *feedforward* konvensional [14, 15]. Lapisan ini umumnya diletakkan di bagian akhir jaringan. Output dari lapisan konvolusi atau pooling terakhir terlebih dahulu diubah menjadi bentuk vektor satu dimensi melalui proses *flatten*, sebelum akhirnya diproses untuk menghasilkan keputusan klasifikasi akhir [14].

2.2.4 Lapisan Non-linearitas (Fungsi Aktivasi)

Fungsi aktivasi berperan menentukan output akhir dari sebuah jaringan saraf, misalnya dalam bentuk keputusan klasifikasi [14]. Fungsi aktivasi dapat dibedakan menjadi dua kategori, yaitu fungsi linear dan fungsi non-linear. Fungsi non-linear lebih banyak digunakan pada jaringan saraf modern karena mampu memodelkan hubungan yang kompleks antara input dan output jaringan, sesuatu yang tidak dapat dilakukan oleh fungsi linear [14].

2.3 *Object Detection*

Object detection merupakan domain fundamental dalam visi komputer yang memiliki tujuan mengidentifikasi kategori entitas serta menentukan lokalisasi spasialnya dalam bentuk kotak pembatas (*bounding box*) [16]. Dibandingkan dengan klasifikasi gambar konvensional, deteksi objek memiliki kompleksitas yang lebih tinggi karena mengharuskan sistem untuk mengeksekusi tugas pengenalan jenis objek dan penentuan posisi presisi dalam ruang gambar secara simultan [17].

Saat ini, efektivitas sistem deteksi objek secara masif didorong oleh pemanfaatan arsitektur *deep learning*, terutama *Convolutional Neural Network* (CNN). Keunggulan CNN terletak pada kemampuannya melakukan ekstraksi fitur hierarkis secara otomatis dari data piksel, yang secara signifikan mengungguli metode tradisional berbasis fitur rancangan manual (*hand-crafter*) yang cenderung kaku [16]. Melalui integrasi operasi konvolusi dan mekanisme regresi, model

ini mampu memprediksi lokasi dan label kelas secara bersamaan [17]. Dalam konteks teknisnya, arsitektur pendeteksi objek berbasis CNN secara garis besar diklasifikasikan ke dalam dua paradigma utama, yaitu *two-stage detector* dan *one-stage detector* [17].

2.3.1 *Two-stage Detector*

Arsitektur *Two-stage detector* bekerja melalui dua tahapan komputasi yang berurutan, yaitu tahap pertama untuk menghasilkan sejumlah usulan wilayah atau *Region of Interest* (ROI) yang diprediksi mengandung objek, serta tahap kedua yang melakukan ekstraksi fitur, klasifikasi, dan regresi untuk menentukan label kelas dan presisi koordinat pembatas (*bounding box*) secara spesifik [16]. Mekanisme ini memberikan keunggulan utama pada standar akurasi lokalisasi dan presisi klasifikasi yang tinggi, sehingga menjadi pilihan utama untuk tugas deteksi yang mengutamakan ketepatan detail [16]. Namun, pemisahan proses ini juga memiliki kelemahan berupa beban komputasi (*overhead*) yang masif akibat evaluasi ribuan kandidat wilayah secara bertahap, yang berimplikasi pada rendahnya kecepatan inferensi model.

2.3.2 *One-stage Detector*

Arsitektur *One-stage detector* beroperasi menggunakan kerangka kerja tunggal yang menggabungkan fungsi regresi dan klasifikasi untuk memproses input gambar secara utuh dalam satu kali tahapan propagasi jaringan tanpa memerlukan ROI secara terpisah [16]. Melalui mekanisme ini, estimasi posisi kotak pembatas (*bounding box*) dan prediksi probabilitas kategori kelas dilakukan secara simultan pada seluruh dimensi spasial gambar [16]. Karakteristik tersebut menghasilkan latensi inferensi yang sangat rendah dengan struktur jaringan yang lebih sederhana, sehingga sangat ideal untuk diimplementasikan pada sistem deteksi *real-time*. Meskipun demikian, arsitektur ini sering kali menunjukkan margin akurasi yang lebih rendah pada beberapa model generasi awal [16].

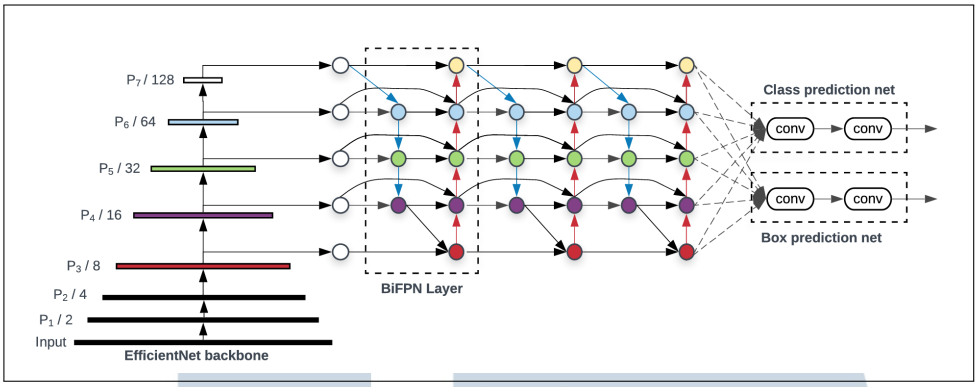
2.4 *Transfer Learning*

Deep Transfer Learning (DTL) merupakan salah satu paradigma pembelajaran mesin yang dikembangkan dengan tujuan menekan biaya pelatihan serta mengurangi ketergantungan pada dataset berlabel dalam jumlah besar, melalui penggunaan kembali pengetahuan yang telah diperoleh dari tugas atau dataset sumber untuk diterapkan pada tugas dataset. Pendekatan ini banyak diadopsi dalam bidang pencitraan medis, mengingat data berlabel pada domain tersebut umumnya tergolong sulit diperoleh [18]. Secara taksonomi, salah satu pendekatan DTL yang paling umum digunakan adalah pendekatan berbasis model atau berbasis parameter. Pada pendekatan ini, model yang telah dilatih sebelumnya (*pre-trained model*) pada dataset berskala besar digunakan sebagai titik awal inisialisasi bobot [18].

2.5 EfficientDet

EfficientDet merupakan keluarga model deteksi objek dengan paradigma *one-stage detector* yang diusulkan oleh Tan, Pang, dan Le dari Google Brain pada tahun 2020 [8]. Model ini dibangun di atas dua inovasi utama, yaitu *Weighted Bidirectional Feature Pyramid Network* (BiFPN) sebagai jaringan fusi fitur dan metode *compound scaling* untuk penskalaan model yang seragam.

Gambar 2.3 menunjukkan arsitektur dari EfficientDet yang terdiri atas tiga komponen utama, yaitu *backbone network*, BiFPN, dan *prediction head*. *Backbone* Arsitektur ini berupa EfficientNet yang telah melalui tahap *pre-training* pada dataset ImageNet. Pada tahap ini, EfficientNet mengekstraksi fitur pada level 3 hingga 7 ($P_3 - P_7$). Selanjutnya, fitur-fitur tersebut digabungkan melalui BiFPN untuk menghasilkan representasi fitur multi-skala yang lebih informatif. Hasil keluaran BiFPN kemudian diteruskan ke *prediction head* yang terdiri atas *class prediction network* dan *box prediction network*. Sebelum menghasilkan prediksi akhir, masing-masing *prediction network* melewati beberapa *repeated convolution layers* untuk memperkaya representasi fitur. Setelah itu, layer konvolusi terakhir menghasilkan prediksi probabilitas kelas objek pada *class prediction network* dan koordinat *bounding box* pada *box prediction network*.



Gambar 2.3. Arsitektur EfficientDet

Sumber: [8]

2.5.1 *Weighted Bidirectional Feature Pyramid Network* (BiFPN)

Weighted Bidirectional Feature Pyramid Network (BiFPN) merupakan komponen penting dalam arsitektur EfficientDet yang berfungsi untuk menggabungkan informasi dari berbagai ukuran gambar secara efisien [8]. Jaringan ini dirancang untuk menyempurnakan metode fusi fitur terdahulu seperti FPN dan PANet yang dinilai masih memiliki keterbatasan dalam hal kecepatan maupun beban komputasi. Inovasi utama pada BiFPN terletak pada kemampuannya untuk memberikan bobot atau nilai kepentingan yang berbeda pada setiap input fitur. Melalui mekanisme ini, model dapat secara otomatis mempelajari fitur mana yang paling penting untuk digunakan dalam proses deteksi [19].

2.5.2 Metode *Compound Scaling*

Metode *compound scaling* merupakan strategi penskalaan model yang digunakan untuk menyeimbangkan tiga dimensi utama jaringan saraf secara bersamaan, yaitu dimensi kedalaman (*depth*), lebar (*width*), dan resolusi input gambar (*resolution*) [8]. Inovasi ini didasarkan pada prinsip bahwa ketiga dimensi tersebut saling bergantung dalam mengoptimalkan efisiensi komputasi model. Sebagai contoh, jika resolusi gambar ditingkatkan, maka jaringan membutuhkan lebih banyak lapisan untuk memperluas bidang reseptif dan lebih banyak saluran untuk menangkap detail pola yang lebih halus. Dengan menerapkan metode ini pada seluruh komponen arsitektur mulai dari *backbone* hingga jaringan fusi fitur, EfficientDet mampu menghasilkan akurasi tinggi dengan jumlah parameter yang jauh lebih sedikit dibandingkan model detektor objek lainnya.

Penerapan metode ini diatur oleh satu nilai utama, yaitu parameter ϕ (phi). Parameter ini berfungsi sebagai pengatur skala gabungan yang secara otomatis menentukan besaran teknis pada model, meliputi lebar saluran BiFPN (W_{bifpn}), jumlah lapisan BiFPN (D_{bifpn}), serta kedalaman jaringan prediksi kelas (D_{class}) [8]. Secara teknis, nilai ϕ berkisar dari 0 hingga 7 yang menjadi dasar klasifikasi varian model EfficientDet. Semakin tinggi nilai ϕ, maka nilai (W_{bifpn}), (D_{bifpn}), dan (D_{class}) akan meningkat secara proporsional agar model mampu menangkap detail objek yang lebih rumit. Karakteristik setiap varian model berdasarkan parameter tersebut disajikan pada Tabel 2.1 berikut.

Tabel 2.1. Konfigurasi Skala Arsitektur EfficientDet (D0 – D7x)

Varian	**Input size**	**Backbone**	**BiFPN**		**Box/class**
Model (ϕ)	R_{input}	**Network**	W_{bifpn}	D_{bifpn}	D_{class}
D0 ($\phi = 0$)	512	B0	64	3	3
D1 ($\phi = 1$)	640	B1	88	4	3
D2 ($\phi = 2$)	768	B2	112	5	3
D3 ($\phi = 3$)	896	B3	160	6	4
D4 ($\phi = 4$)	1024	B4	224	7	4
D5 ($\phi = 5$)	1280	B5	288	7	4
D6 ($\phi = 6$)	1280	B6	384	8	5
D7 ($\phi = 7$)	1536	B6	384	8	5
D7x	1536	B7	384	8	5

2.5.3 EfficientDet-D1

EfficientDet-D1 adalah salah satu varian dalam keluarga EfficientDet yang dihasilkan melalui penerapan *compound scaling*. EfficientDet-D1 menggunakan *backbone* EfficientNet-B1 dengan resolusi input gambar sebesar 640×640 piksel. *Backbone* ini menghasilkan *feature map* pada beberapa tingkatan skala yang selanjutnya diteruskan menuju jaringan BiFPN untuk dilakukan proses fusi fitur secara dua arah.

Feature map hasil keluaran BiFPN kemudian diteruskan menuju *prediction head* yang terdiri atas dua jaringan, yaitu *class prediction network* dan *box prediction network*. Masing-masing prediction network terdiri atas tiga *repeated convolution layers* sebelum diproses oleh layer konvolusi terakhir untuk

menghasilkan prediksi [8]. Konfigurasi utama EfficientDet-D1 dapat dilihat pada Tabel 2.2.

Tabel 2.2. Konfigurasi EfficientDet-D1

Komponen	**Konfigurasi EfficientDet-D1**
Backbone	EfficientNet-B1
Resolusi Input	640×640 piksel
Jumlah Kanal BiFPN	88
Jumlah Lapisan BiFPN	4
Jumlah Layer *Prediction Head*	3

2.6 *Fine-Tuning*

Fine-tuning merupakan salah satu teknik utama dalam DTL, yaitu proses menyesuaikan *parameter* atau bobot dari model *pre-trained* pada tugas target [18]. Berdasarkan strategi penyesuaian parameternya, teknik *fine-tuning* dibagi menjadi beberapa varian berikut.

1. *Full Fine-tuning*
   Pada pendekatan ini, seluruh parameter model diperbarui selama proses pelatihan pada data target. Meskipun tergolong sederhana secara konsep, teknik ini umumnya diasosiasikan dengan beban komputasi yang tinggi serta risiko melupakan informasi dari bobot data sebelumnya [20].

2. *Partial Fine-tuning (Freezing Layers)*
   Strategi ini dilakukan dengan membekukan lapisan awal dan menengah, yang umumnya berfungsi mengekstraksi fitur umum, sementara penyesuaian parameter hanya dilakukan pada lapisan lateral atau akhir yang dianggap lebih spesifik terhadap klasifikasi tugas target [18].

3. *Linear Probe*
   Linear probe merupakan salah satu bentuk spesifik dari pembekuan lapisan, yaitu penambahan sebuah lapisan linear baru sebagai lapisan klasifikasi di atas model *pre-trained*, sementara seluruh parameter model dasar tetap dibekukan atau tidak diperbarui [20].

2.7 *Catastrophic Forgetting*

Catastrophic Forgetting adalah kondisi ketika pengetahuan yang telah diperoleh dari dataset sumber terhapus sebagian atau seluruhnya pada saat model mempelajari informasi baru dari tugas target [18]. Guna mengurangi risiko tersebut dalam praktik *fine-tuning*, terdapat beberapa strategi yang dapat diterapkan yaitu sebagai berikut.

1. Analisis *Module Criticality*
   Tingkat kepentingan atau kritikalitas suatu lapisan meningkat seiring pergerakan dari sisi input (lapisan awal) menuju sisi output (lapisan akhir). Oleh karena itu, penyesuaian parameter perlu dilakukan secara lebih hati-hati pada lapisan akhir, mengingat fiturnya lebih terspesialisasi terhadap domain target, sementara lapisan awal bersifat lebih umum dan stabil [18].

2. Pengaturan *Learning Rate*
   Kinerja model *fine-tuning* cukup sensitif terhadap konfigurasi *learning rate*. Penyesuaian yang kurang tepat pada parameter pelatihan dapat menyebabkan variasi performa yang signifikan serta memicu *catastrophic forgetting* [20]. Strategi yang menjaga jaringan tetap berada dalam basin solusi yang sama dengan model *pre-trained* dapat membantu menjaga stabilitas pengetahuan selama proses penyesuaian [18].

2.8 Teknik *Windowing Hounsfield Unit*

Windowing merupakan teknik penyesuaian rentang intensitas piksel pada citra medis berskala *Hounsfield Unit* (HU) untuk menonjolkan visualisasi struktur anatomi tertentu secara spesifik [6]. Secara matematis, teknik ini membatasi rentang nilai piksel menggunakan dua parameter utama, yakni titk tengah jendela (c_W) dan lebar jendela (w_W). Batas atas (max_W) dan batas bawah (min_W) dari rentang jendela tersebut dihitung melalui persamaan yang dapat dilihat pada Rumus 2.1 dan 2.2.

$$max_W = c_W + \frac{w_W}{2} \tag{2.1}$$

$$min_W = c_W - \frac{w_W}{2} \tag{2.2}$$

Seluruh nilai intensitas HU yang berada di luar batas tersebut akan dipotong. Nilai yang tersisa di dalam jendela kemudian dinormalisasi menjadi format citra standar 8-bit dengan rentang piksel 0 hingga 255 menggunakan Rumus 2.3 [6].

$$p = \frac{HU - min_W}{max_W - min_W} \times 255 \tag{2.3}$$

2.9 *Confusion Matrix*

Confusion matrix merupakan alat evaluasi yang digunakan untuk merepresentasikan dan merangkum kinerja model klasifikasi. Evaluasi ini didasarkan pada perbandingan silang antara kondisi aktual data dengan kondisi hasil prediksi yang dikeluarkan oleh algoritma. Terdapat empat kategori luaran prediksi dalam matriks ini. *True Positive* (TP) terjadi apabila suatu sampel secara aktual bernilai positif dan model juga memprediksinya dengan benar sebagai positif. *True Negative* (TN) terjadi apabila sampel secara aktual bernilai negatif dan model memprediksinya dengan benar sebagai negatif. Sebaliknya, luaran prediksi keliru terbagi menjadi dua, yakni *False Positive* (FP) yang terjadi saat sampel aktual bernilai negatif namun diprediksi secara keliru sebagai positif, serta *False Negative* (FN) yang terjadi saat sampel aktual bernilai positif namun diprediksi secara keliru sebagai negatif [21].

2.10 *Precision* dan *Recall*

Precision didefinisikan sebagai kemampuan model dalam mengidentifikasi hanya objek yang benar-benar relevan dari keseluruhan hasil prediksi yang dilabeli positif oleh sistem [22]. Secara matematis, *precision* dihitung dengan membagi jumlah prediksi benar (*True Positive*) dengan total keseluruhan prediksi positif (kombinasi *True Positive* dan *False Positive*) [21]. Rumus *precision* didefinisikan pada Rumus 2.4.

$$Precision = \frac{TP}{TP + FP} \tag{2.4}$$

Sementara itu, *recall* mengukur kemampuan model dalam menemukan atau mengenali keseluruhan kasus positif yang sebenarnya ada pada data aktual [22].

Nilai *recall* diperoleh dengan membagi jumlah prediksi benar (*True Positive*) dengan total data aktual yang bernilai positif (kombinasi *True Positive* dan *False Negative*) [21]. Rumus *recall* didefinisikan pada Rumus 2.5.

$$Recall = \frac{TP}{TP + FN} \tag{2.5}$$

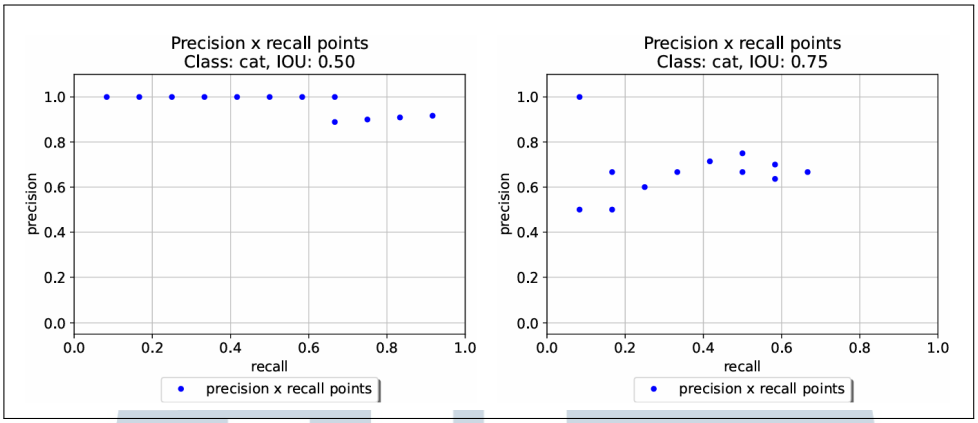
2.11 *F1-Score*

F1-Score adalah nilai keseimbangan antara *precision* dan *recall*. Metrik ini sering digunakan sebagai indikator tunggal yang paling adil untuk melihat performa model, terutama jika data yang digunakan tidak seimbang jumlahnya antara satu kelas dengan kelas lainnya [23]. Jika nilai *F1-Score* tinggi, berarti model memiliki performa yang stabil antara ketepatan hasil dan cakupan deteksi [24]. Rumus 2.6 menunjukkan cara perhitungan dari *F1-Score*.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.6}$$

2.12 *Mean Average Precision* (mAP)

Dalam tugas deteksi objek, kinerja model dievaluasi menggunakan metrik *mean Average Precision* (mAP), yang merupakan nilai rata-rata dari *Average Precision* (AP) di seluruh kelas yang diuji. Secara fundamental, AP merepresentasikan luas area di bawah kurva *Precision-Recall* (AUC) [21]. Kurva ini dibentuk dengan nilai *precision* pada sumbu Y terhadap nilai *recall* pada sumbu X untuk berbagai tingkat ambang batas kepercayaan (*confidence threshold*) [22]. Karakteristik kurva *precision-recall* sering kali menunjukkan perilaku ”zig-zag” atau tidak monoton sebagaimana ditunjukkan pada Gambar 2.4.



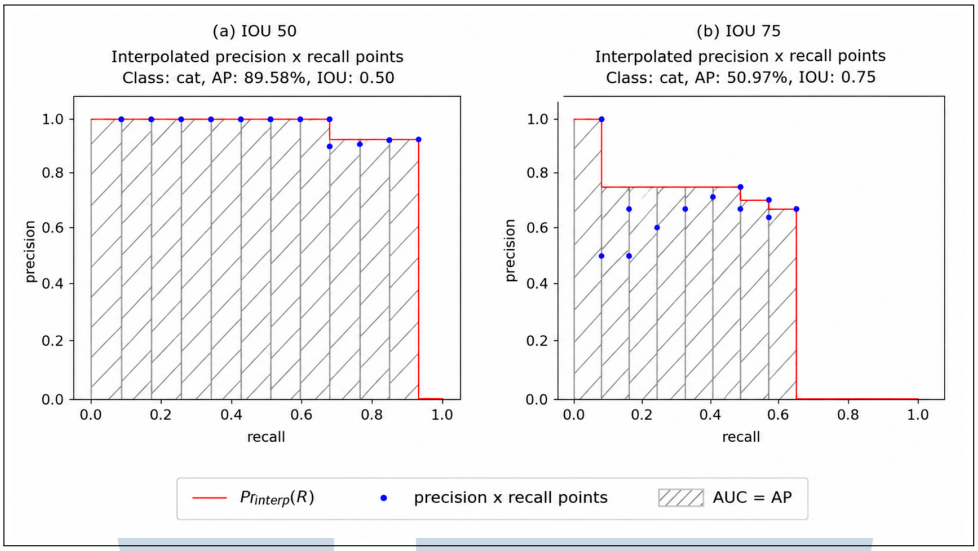
Gambar 2.4. Kurva Precision-Recall

Sumber: [22]

Berdasarkan ambang batas *Intersection over Union* (IoU) tertentu (u), nilai AP_u didefinisikan sebagai integral dari fungsi presisi (p_u) terhadap recall (r) dalam rentang 0 hingga 1 yang dapat dilihat pada Rumus 2.7.

$$AP_u = \int_0^1 p_u(r)\,dr \tag{2.7}$$

Namun, karena perilaku ”zig-zag” pada kurva mentah menyulitkan perhitungan luas area secara akurat, nilai AP dalam praktiknya dhitung menggunakan *interpolated precision* (P_{interp}) untuk menghaluskan kurva. Karakteristik kurva yanng telah melalui proses interpolasi ditunjukkan pada Gambar 2.5.



Gambar 2.5. Kurva Interpolated Precision-Recall

Sumber: [22]

Nilai *interpolated precision* pada tingkat recall R didefinisikan sebagai nilai presisi maksimum untuk setiap tingkat *recall* yang lebih besar atau sama dengan R. Rumus *interpolated precision* dapat dilihat pada Rumus 2.8.

$$P_{interp}(R) = \max_{k|Rc(\tau(k))\geq R} \{Pr(\tau(k))\} \tag{2.8}$$

Penentuan nilai *precision* dan *recall* tersebut sangat bergantung pada ambang batas IoU (u) yang ditetapkan. Jika ambang batas IoU dinaikkan, maka kriteria model untuk menganggap sebuah deteksi sebagai benar (*True Positive*) menjadi lebih sulit dipenuhi [22]. Hal ini mengakibatkan terjadinya penurunan nilai *precision* dan *recall* secara bersamaan karena kotak prediksi yang memiliki lokalisasi dibawah ambang batas akan diklasifikasikan sebagai *False Positive*. Berdasarkan variasi ambang batas IoU tesebut, terdapat dua varian mAP yang umum digunakan sebagai standar evaluasi yaitu sebagai berikut.

1. mAP_{50}

   Metrik mAP_{50} adalah nilai mAP yang dihitung secara spesifik menggunakan ambang batas IoU yang moderat, yakni sebesar 50% atau 0,50 [25]. Variabel N merepresentasikan jumlah total kelas yang dievaluasi, sedangkan $AP_{50,i}$ merepresentasikan nilai *Average Precision* pada kelas ke-i dengan menggunakan ambang batas IoU 50%. Rumus untuk mAP_{50} didefinisikan pada Rumus 2.9.

$$mAP_{50} = \frac{1}{N}\sum_{i=1}^{N} AP_{50,i} \tag{2.9}$$

2. $mAP_{50:95}$

   $mAP_{50:95}$ merupakan metrik evaluasi yang jauh lebih komprehensif karena metrik ini menghitung rata-rata nilai mAP yang diuji pada sepuluh tingkat ambang batas IoU yang berbeda-beda, dimulai dari IoU 50% hingga 95% dengan interval kenaikan sebesar 5%. Persamaan matematis untuk $mAP_{50:95}$ dijabarkan pada Rumus 2.10 [25].

$$mAP_{50:95} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{10}\sum_{j=1}^{10} AP_{50+5(j-1),i}\right) \tag{2.10}$$